

# Counting perfect matchings in graphs that exclude a single-crossing minor

Radu Curticapean\*

## Abstract

A graph  $H$  is *single-crossing* if it can be drawn in the plane with at most one crossing. For any single-crossing graph  $H$ , we give an  $\mathcal{O}(n^4)$  time algorithm for counting perfect matchings in graphs excluding  $H$  as a minor. The runtime can be lowered to  $\mathcal{O}(n^{1.5})$  when  $G$  excludes  $K_5$  or  $K_{3,3}$  as a minor.

This is the first generalization of an algorithm for counting perfect matchings in  $K_{3,3}$ -free graphs (Little 1974, Vazirani 1989). Our algorithm uses black-boxes for counting perfect matchings in planar graphs and for computing certain graph decompositions. Together with an independent recent result (Straub et al. 2014) for graphs excluding  $K_5$ , it is one of the first nontrivial algorithms to not inherently rely on Pfaffian orientations.

## 1 Introduction

A *perfect matching* of a graph  $G = (V, E)$  is a set  $M \subseteq E$  of  $|V|/2$  vertex-disjoint edges. For an edge-weighted graph  $G$  with weights  $w : E \rightarrow \mathbb{Q}$ , we consider the problem of computing  $\text{PerfMatch}(G) = \sum_M \prod_{e \in M} w(e)$ , where the outer sum ranges over all perfect matchings  $M$  of  $G$ . If  $w(e) = 1$  for all  $e \in E(G)$ , this quantity plainly counts perfect matchings of  $G$ .

The problem  $\text{PerfMatch}$  arises in statistical physics as the dimer problem [9, 17]. In algebra and combinatorics, the quantity  $\text{PerfMatch}(G)$  for bipartite  $G$  is better known as the permanent of the (bi-)adjacency matrix of  $G$ . The complexity of its evaluation is of central interest in counting complexity [18] and algebraic complexity [3]. In fact, the permanent was the first natural problem with a polynomial-time decision version that was shown  $\#\text{P}$ -hard, even for zero-one weights, thus demonstrating that counting can be harder than decision.

To cope with this hardness, several reliefs were proposed: If counting may be relaxed to approximate counting, then the problem becomes feasible: It was shown in [8] that  $\text{PerfMatch}(G)$  admits a fully polynomial randomized approximation scheme on graphs  $G$  with non-negative edge weights. If the exact value of  $\text{PerfMatch}(G)$  is required, but  $G$  may be restricted to a specific class of graphs, then a rather short list of polynomial-time algorithms is known:

For planar  $G$ , the value  $\text{PerfMatch}(G)$  can be computed in time  $\mathcal{O}(n^{1.5})$  by [17, 9]. Interestingly, this algorithm from 1967 predates the hardness result for general graphs. Note that planar graphs exclude both  $K_{3,3}$  and  $K_5$  as a minor. In [12, 20], the previous algorithm was generalized to a (parallel) algorithm on graphs  $G$  that are only required to exclude the minor  $K_{3,3}$ . Orthogonally to this, it was shown in [7] that  $\text{PerfMatch}(G)$  admits an  $\mathcal{O}(4^g n^3)$

---

\*Saarland University, Dept. of Computer Science, [curticapean@cs.uni-sb.de](mailto:curticapean@cs.uni-sb.de)

algorithm on graphs that can be embedded on a surface of genus  $g$ . Recently, and independently of this work, a (parallel) polynomial-time algorithm was shown in [16] for computing  $\text{PerfMatch}(G)$  on graphs excluding  $K_5$  as a minor. In the present paper, we show:

**Theorem 1.** *Let  $H$  be a single-crossing graph, i.e.,  $H$  can be drawn in the plane with at most one crossing. Then there is an  $\mathcal{O}(n^4)$  time algorithm for computing  $\text{PerfMatch}(G)$  on input graphs  $G$  that exclude  $H$  as a minor. If  $H$  is one of the single-crossing graphs  $K_5$  or  $K_{3,3}$ , then the runtime can be lowered to  $\mathcal{O}(n^{1.5})$ .*

Note that the excluded minor  $H$ , rather than  $G$ , is required to be single-crossing: Algorithms for single-crossing  $G$  would follow from a very simple reduction to the planar case.

Theorem 1 directly generalizes the algorithm for graphs excluding  $K_{3,3}$  or  $K_5$ , but is orthogonal to the result for bounded-genus graphs: The graph consisting of  $n$  disjoint copies of the single-crossing graph  $K_5$  has genus  $\Theta(n)$ , but excludes  $K_{3,3}$  as a minor. Thus, Theorem 1 applies on this graph, while the algorithm for bounded-genus graphs does not. Conversely, the class of torus-embeddable graphs includes all single-crossing graphs. Thus, the algorithm for bounded-genus graphs applies here, while Theorem 1 does not.

Graphs excluding a single-crossing minor  $H$  have already been studied: By a decomposition theorem [14], which constitutes a fragment of the general graph structure theorem for general  $H$ -minor free graphs [15], such graphs can be decomposed into planar graphs and graphs of bounded treewidth, and it was shown in [5] how to compute such decompositions. Furthermore, approximation algorithms for the treewidth and other invariants of such graphs are known [5, 6], as well as  $\mathcal{O}(n \log n)$  algorithms for computing maximum flows [4].

Our algorithm requires black-boxes for  $\text{PerfMatch}$  on planar graphs and for finding the decompositions described above. We also use the concept of matchgates from [19], but can limit ourselves to a self-contained fragment of their theory. All required ingredients are introduced in Section 2 and used in Section 3 to present the algorithm proving Theorem 1.

## 2 Mise en place

Let  $\mathbb{F}$  be a field supporting efficient arithmetic operations. Graphs  $G = (V, E)$  are undirected and may feature parallel edges and weights  $w : E \rightarrow \mathbb{F}$ . We allow zero-weight edges  $e \in E$  with  $w(e) = 0$  and write  $|G| := |V(G)|$ .

A graph  $G$  is planar if it admits an embedding  $\pi$  into the plane without crossings, and single-crossing if it admits an embedding into the plane with at most one crossing. Examples for single-crossing graphs are  $K_5$  and  $K_{3,3}$ . A plane graph is a pair  $(G, \pi)$ , where  $\pi$  is a planar embedding of  $G$ . Given a plane graph  $(G, \pi)$  and a cycle  $C$  in  $G$ , we say that  $C$  bounds a face in  $G$  if one of the regions bounded by  $C$  in  $\pi$  is empty.

We write  $\mathcal{PM}[G]$  for the set of perfect matchings of  $G$  and define  $w(M) = \prod_{e \in M} w_G(e)$  and  $\text{PerfMatch}(G) = \sum_{M \in \mathcal{PM}[G]} w(M)$ . As already noted, despite its  $\#P$ -hardness on general graphs, the value  $\text{PerfMatch}(G)$  can be computed in polynomial time for planar  $G$ .

**Theorem 2.** *For planar graphs  $G$ , the value  $\text{PerfMatch}(G)$  can be computed in time  $\mathcal{O}(n^{1.5})$ .*

*Proof.* (Sketch of [9]) In time  $\mathcal{O}(n)$ , we can compute a set  $S \subseteq E(G)$  such that the following holds: After flipping the sign of  $w(e)$  for each edge  $e \in S$ , we obtain a new planar graph with adjacency matrix  $A'$  satisfying  $\text{PerfMatch}(G) = \sqrt{\det(A')}$ . If  $A'$  is the adjacency matrix of a planar graph, then  $\det(A')$  can be computed in time  $\mathcal{O}(n^{1.5})$  by [11], noted also in [19].  $\square$

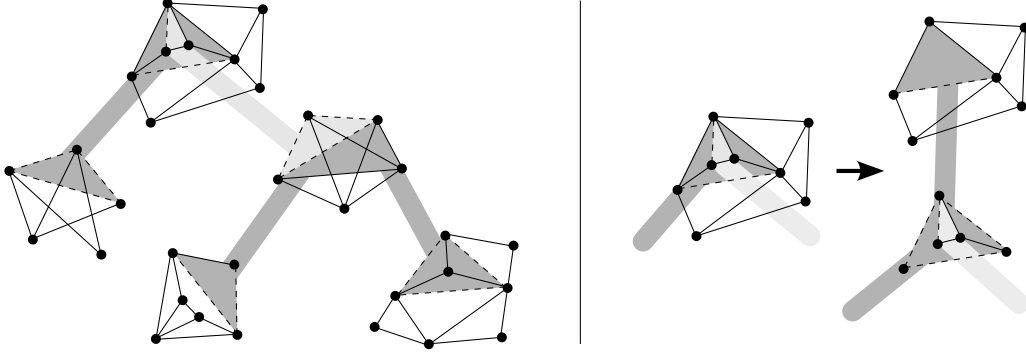


Figure 1: (left)  $\mathcal{T}$  is almost 5-nice: Either  $|V(G_t)| \leq 5$  or  $G_t$  is a plane graph whose non-navel attachment cliques bound faces, with the exception of one triangle  $K$  at the root. Zero-weight edges are drawn with dashed lines. (right) The offending attachment clique  $K$  is repaired.

## 2.1 Graph minors and decompositions

A graph  $H$  is a minor of  $G = (V, E)$  if  $H$  can be obtained from  $G$  by repeated edge/vertex-deletions and edge-contractions. The contraction of  $uv \in E$  identifies vertices  $u, v \in V(G)$  to a new vertex  $w$  and replaces possible edges  $uz \in E$  or  $vz \in E$  for  $z \in V(G)$  by a new edge  $wz$ . For a graph class  $\mathcal{H}$ , write  $\mathcal{C}[\mathcal{H}]$  for the class of all graphs  $G$  such that no  $H \in \mathcal{H}$  is a minor of  $G$ . By Kuratowski's theorem,  $\mathcal{C}[K_{3,3}, K_5]$  coincides with the planar graphs.

Other graph classes can also be expressed by forbidden minors. In fact, Robertson and Seymour's graph structure theorem [15] describes the structure of graphs in  $\mathcal{C}[H]$  for arbitrary  $H$ . We use a fragment of this theorem that applies only when  $H$  is single-crossing: Roughly speaking, graphs in  $\mathcal{C}[H]$  consist of planar graphs and constant-size graphs that are glued together in a well-specified way. Our algorithm will crucially rely on these decompositions.

**Definition 1.** Let  $F, F'$  be graphs, both containing a vertex set  $K$ . Write  $F \oplus_K F'$  for the graph obtained from the disjoint union of  $F$  and  $F'$  by identifying, for each  $v \in K$ , the two copies of  $v$ . This may create parallel edges between vertices in  $K$ .

- In the following, let  $G$  be a graph. A *decomposition*  $\mathcal{T} = (T, \mathcal{G})$  of  $G$  is a rooted tree  $T$  with a family of graphs  $\mathcal{G} = \{G_t\}_{t \in V(T)}$  such that the following holds:
  1. For  $st \in E(T)$ , the set  $K[s, t] := V(G_s) \cap V(G_t)$  is a clique, the so-called *attachment clique* at  $st$ , possibly containing zero-weight edges in  $G_s$  or  $G_t$ . If  $s$  is the parent of  $t$ , we call  $K[s, t]$  the *navel* of  $t$ .
  2. For  $t \in V(T)$ , define  $G_{\leq t}$ : If  $t$  is a leaf, then  $G_{\leq t} = G_t$ . If  $t$  has children  $s_1, \dots, s_r$  with navels  $K_1, \dots, K_r$ , then  $G_{\leq t} = G_t \oplus_{K_1} G_{\leq s_1} \oplus_{K_2} \dots \oplus_{K_r} G_{\leq s_r}$ . If  $t$  is the root, we require that  $G_{\leq t}$  is isomorphic to  $G$  after removal of all zero-weight edges.
- For  $c \in \mathbb{N}$ , the decomposition  $\mathcal{T}$  is *c-nice* if  $G_t$  is given as a plane graph whenever  $|V(G_t)| > c$ . Furthermore, if  $K$  is an attachment clique in  $G_t$ , then  $|K| \leq 3$ . If  $|K| = 3$  and  $K$  is not the navel of  $G_t$ , then  $K$  is required to bound a face in  $G_t$ .
- If  $|V(G_t)| \leq k$  for all  $t \in V(T)$ , then  $\mathcal{T}$  is a *tree-decomposition* of width  $k$  of  $G$ . The *treewidth* of  $G$  is defined as  $\min\{k \in \mathbb{N} \mid G \text{ has a tree-decomposition of width } k + 1\}$ .

*Remark 1.* The above definition of treewidth, used e.g. in [10], is equivalent to the more common one that uses “bags”. It is also verified that, if  $\mathcal{T}$  is a decomposition of  $G$  and  $K$  is a clique in  $G$ , then there is some node  $t$  in  $\mathcal{T}$  such that  $K \subseteq V(G_t)$ .

**Theorem 3.** *For every single-crossing graph  $H$ , there is a constant  $c \in \mathbb{N}$  such that the following holds: For every  $G \in \mathcal{C}[H]$ , a  $c$ -nice decomposition  $\mathcal{T} = (T, \mathcal{G})$  of  $G$  can be found in time  $\mathcal{O}(n^4)$ . Additionally,  $\mathcal{T}$  satisfies the size bounds  $\sum_{t \in V(T)} |G_t| \in \mathcal{O}(n)$  and  $|T| \in \mathcal{O}(n)$ .*

*Proof.* Using the decomposition algorithm presented in [5], we compute in  $\mathcal{O}(n^4)$  time a decomposition  $\mathcal{T}' = (T', \mathcal{G}')$  that satisfies the following: For each  $t \in V(T')$ , either  $G_t$  has treewidth  $\leq c$ , or  $G_t$  is a plane graph whose attachment cliques  $K$  satisfy  $|K| \leq 3$ . Furthermore,  $\mathcal{T}'$  satisfies the size bounds stated in the theorem for  $\mathcal{T}$ .

By local patches at nodes  $t \in V(T)$ , we successively transform  $\mathcal{T}'$  to a  $c$ -nice decomposition  $\mathcal{T}$ . This involves (i) splitting nodes  $t$  of treewidth  $\leq c$  into trees of constant-size parts, and (ii) splitting planar nodes into multiple planar nodes whose non-navel attachments bound faces.

With  $Z_t$  denoting the set of nodes added to  $\mathcal{T}'$  by patching  $t$ , we show along the way that the local size bound  $\sum_{z \in Z_t} |G_z| \in \mathcal{O}(|G_t|)$  holds. This implies the claimed size bounds on  $\mathcal{T}$ .

(i) Let  $G_t$  have treewidth  $\leq c$ . Using [2], compute in time  $\mathcal{O}(2^{c^3}n)$  a tree-decomposition  $\mathcal{R} = (R, \mathcal{B})$  of width  $c$  of  $G_t$  with  $\mathcal{B} = \{B_r\}_{r \in V(R)}$  and  $|R| \in \mathcal{O}(|G_t|)$ . Let  $K$  be the navel of  $t$  and let  $r$  be an arbitrary node of  $R$  satisfying  $K \subseteq V(B_r)$ , which exists by Remark 1. Declare  $r$  as root of  $\mathcal{R}$  and attach  $\mathcal{R}$  to  $\mathcal{T}'$  by deleting  $t$  from  $\mathcal{T}'$ , disconnecting possible children of  $t$ , and inserting  $\mathcal{R}$  with root  $r$  at the place of  $t$ . For every child  $s$  of  $t$  in  $\mathcal{T}'$  that was disconnected this way, do the following: By Remark 1, its navel, which is a clique, is contained in  $B_p$  for some node  $p$  of  $\mathcal{R}$ . Add the edge  $ps$  to  $\mathcal{T}'$ . Processing  $t$  this way adds  $|R| \in \mathcal{O}(|G_t|)$  new nodes  $z$  to  $\mathcal{T}'$ , each with  $|G_z| \leq c$ , showing the local size bound for  $t$ .

(ii) Similar to [4]. Let  $K$  be an attachment clique of  $G_t$  that does not bound a face, as in Figure 1. Then  $t$  has a neighbor  $s$  such that the subgraph  $F$  bounded by  $K = K[s, t]$  in the embedding of  $G_t$  contains other vertices than  $K$ . Delete  $F - K$  from  $G_t$ . Add a new node  $t'$  adjacent to  $t$  and define  $G_{t'} := F$  with zero weight at all edges in  $F[K]$ . For each child  $r$  of  $t$  whose navel is contained in  $V(F)$ , replace the edge  $rt$  of  $T$  by  $rt'$ . If the newly created graph  $G_{t'}$  contains another attachment clique that does not bound a face, recurse on  $G_{t'}$ .

For (ii), we see that  $|Z_t| \leq |G_t|$  since every recursion step deletes at least one vertex from its current subgraph of  $G_t$ . Secondly, the local size bound holds at  $t$  since every recursion step introduces at most 3 new vertices, namely the copy of  $K$  in the child node.  $\square$

*Remark 2.* For  $H \in \{K_{3,3}, K_5\}$ , an  $\mathcal{O}(1)$ -nice decomposition  $\mathcal{T}$  can be found in time  $\mathcal{O}(n)$ : Instead of computing  $\mathcal{T}'$  by [5] in the first step, use [1] for  $H = K_{3,3}$  or [13] for  $H = K_5$ .

## 2.2 Matchgates and signatures

In the following, we present the concept of matchgates from [19], as these will play a central role in our algorithm. We limit ourselves to a small self-contained fragment of their theory.

**Definition 2** ([19]). A *matchgate*  $\Gamma = (G, S)$  is a graph  $G$  with a set of external vertices  $S \subseteq V(G)$ . Its *signature*  $\text{Sig}(\Gamma) : 2^S \rightarrow \mathbb{F}$  is the function that maps  $X \subseteq S$  to  $\text{PerfMatch}(G - X)$ .

*Remark 3.* For  $\Gamma = (G, S)$  with  $|S| = k$ , we represent  $\text{Sig}(\Gamma)$  by a vector in  $\mathbb{F}^{2^k}$ . If we can compute  $\text{PerfMatch}(G - X)$  for  $X \subseteq S$  in time  $t$ , then we can compute  $\text{Sig}(\Gamma)$  in time  $\mathcal{O}(2^k t)$ .

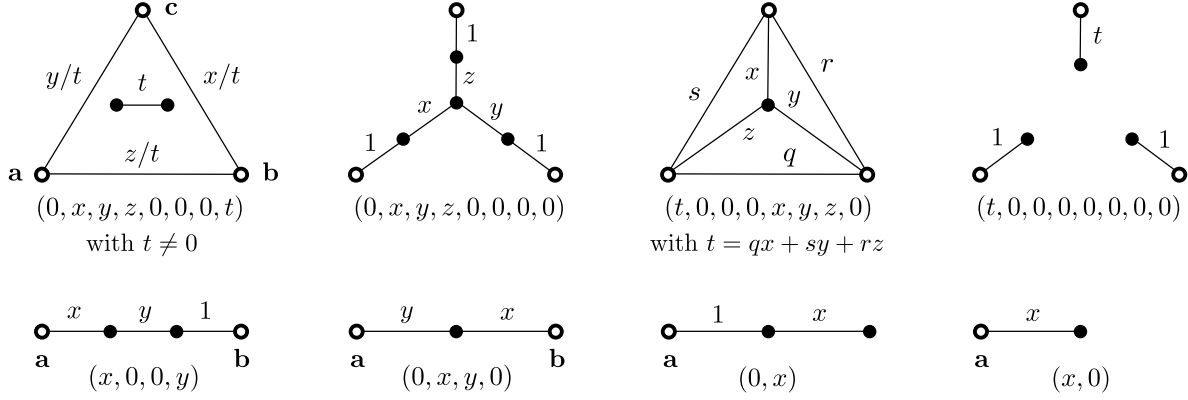


Figure 2: The matchgates from Propositions 6.1 and 6.2 in [19], each drawn as a plane graph with a set  $S \subseteq \{\mathbf{a}, \mathbf{b}, \mathbf{c}\}$  as external vertices on the outer face. Below each matchgate, its signature is given as a vector of length  $2^{|S|}$  with entries ordered as  $\emptyset, a, b, c, ab, ac, bc, abc$  or a subsequence thereof. If  $f$  is even or odd, then at least one matchgate  $\Gamma$  satisfies  $\text{Sig}(\Gamma) = f$ : If  $|S| = 3$  and  $f$  is even, then either the first or second matchgate applies. If  $|S| = 3$  and  $f$  is odd, the third or fourth matchgate applies. If  $|S| \leq 2$ , a matchgate of the second row applies.

The signature of  $\Gamma$  describes its behavior in sums with other graphs:

**Lemma 1.** *For matchgates  $\Gamma = (G, S)$  and  $\Gamma' = (G', S)$ , let  $G^* = G \oplus_S G'$ . Then*

$$\text{PerfMatch}(G^*) = \sum_{Y \subseteq S} \text{Sig}(\Gamma, Y) \cdot \text{Sig}(\Gamma', S \setminus Y). \quad (1)$$

*Proof.* Each  $M \in \mathcal{PM}[G^*]$  induces a unique partition into  $M = N \cup N'$  with  $N \subseteq E(G)$  and  $N' \subseteq E(G')$ . Since  $M$  is a perfect matching, every  $v \in V(G^*)$  is matched in exactly one of  $N$  or  $N'$ . For vertices  $v \notin S$ , the choice of  $N$  or  $N'$  is independent of  $M$ .

For  $Y \subseteq S$ , let  $\mathcal{M}_Y \subseteq \mathcal{PM}[G^*]$  denote the perfect matchings of  $G^*$  with  $S \setminus Y$  matched by  $N$  and  $Y$  matched by  $N'$ . Since  $\{\mathcal{M}_Y\}_{Y \subseteq S}$  partitions  $\mathcal{PM}[G^*]$ , we have  $\text{PerfMatch}(G^*) = \sum_{Y \subseteq S} \sum_{M \in \mathcal{M}_Y} w(M)$ . It remains to show  $\sum_{M \in \mathcal{M}_Y} w(M) = \text{Sig}(\Gamma, Y) \cdot \text{Sig}(\Gamma', S \setminus Y)$ : This follows since every  $M \in \mathcal{M}_Y$  can be written as  $M = N \cup N'$  with  $(N, N') \in \mathcal{PM}[G - Y] \times \mathcal{PM}[G' - (S \setminus Y)]$  and the correspondence between  $M$  and  $(N, N')$  is bijective.  $\square$

Since the only information used about  $G'$  in (1) is contained in  $\text{Sig}(\Gamma')$ , we conclude:

**Corollary 1.** *Let  $\Gamma = (F, S)$  and  $\Gamma' = (F', S)$  and let  $G$  be a graph with  $S \subseteq V(G)$ . If  $\text{Sig}(\Gamma) = \text{Sig}(\Gamma')$ , then  $\text{PerfMatch}(G \oplus_S \Gamma) = \text{PerfMatch}(G \oplus_S \Gamma')$ .*

Whenever  $\Gamma$  has  $\leq 3$  external vertices, we can find a small planar matchgate  $\Gamma'$  with the same signature. We show this in the next fact, essentially from [19]. Together with Corollary 1, we will use  $\Gamma'$  to mimic  $\Gamma$ , similarly to an idea in [4] for mimicking flow networks.

**Fact 1.** *For every matchgate  $\Gamma = (G, S)$  with  $|S| \leq 3$ , there is a matchgate  $\Gamma' = (F, S)$  with  $\text{Sig}(\Gamma) = \text{Sig}(\Gamma')$  such that  $F$  is a plane graph on  $\leq 7$  vertices with  $S$  on its outer face.*

*Proof.* We call  $f : 2^S \rightarrow \mathbb{F}$  even if  $f(X) = 0$  for all  $X$  of odd cardinality, and we call  $f$  odd if  $f(X) = 0$  for all  $X$  of even cardinality. Since every matching features an even number of matched vertices,  $\text{Sig}(\Gamma)$  is even/odd if  $|G|$  is even/odd. Hence Figure 2, adapted from [19], contains a matchgate with signature  $\text{Sig}(\Gamma)$  after suitable substitution of edge weights.  $\square$

### 3 Proof of Theorem 1

By Theorem 3, if  $G$  excludes a fixed single-crossing minor  $H$ , we can find a  $c$ -nice decomposition  $\mathcal{T} = (T, \mathcal{G})$  with  $c \in \mathcal{O}(1)$ . This  $\mathcal{T}$  satisfies  $\sum_{t \in V(T)} |G_t| \in \mathcal{O}(n)$  and  $|T| \in \mathcal{O}(n)$ .

For  $t \in V(T)$ , let  $n_t = |G_t|$ . For non-root nodes  $t \in V(T)$  with navel  $K$ , define the matchgate  $\Gamma_{\leq t} = (G_{\leq t}, K)$ . For the root  $r \in V(T)$ , note that  $G_{\leq r} = G$ . Since  $r$  has no navel, write  $\Gamma_{\leq r} = (G, \emptyset)$  by convention.

We compute  $\text{Sig}(\Gamma_{\leq t})$  for each  $t \in V(T)$  by a bottom-up traversal of  $\mathcal{T}$ . This computes  $\text{Sig}(\Gamma_{\leq r}, \emptyset)$  for the root  $r$ , which is equal to  $\text{PerfMatch}(G)$  by definition. To process  $t \in V(T)$ , we assume that  $\text{Sig}(\Gamma_{\leq r})$  is known for each child  $r$  of  $t$ . This is trivially true if  $t$  is a leaf and will be assumed by induction for non-leaf nodes. We then compute  $\text{Sig}(\Gamma_{\leq t})$  as follows:

- If  $G_t$  has  $\leq c$  vertices, let  $V = V(G_t)$ , let  $\Delta_0 = (G_t, V)$  and compute  $\text{Sig}(\Delta_0)$  in time  $2^{\mathcal{O}(c^2)}$  by brute force. Let  $s_1, \dots, s_b$  be the children of  $t$ , with navels  $K_1, \dots, K_b \subseteq V$ . For  $1 \leq i \leq b$ , define  $\Delta_i = (G_t \oplus_{K_1} G_{\leq s_1} \oplus_{K_2} \dots \oplus_{K_i} G_{\leq s_i}, V)$  and successively compute  $\text{Sig}(\Delta_i)$  from the values of  $\text{Sig}(\Delta_{i-1})$  and  $\text{Sig}(G_{\leq s_i})$  by means of Lemma 1 and Remark 3. After completing this, since the external nodes  $V$  of  $\Delta_b$  trivially include the navel of  $t$ , we obtain  $\text{Sig}(\Gamma_{\leq t})$  as a restriction of  $\text{Sig}(\Delta_b)$ .
- If  $G_t$  is planar, first perform the following for each attachment clique  $K$  of  $G_t$ :
  1. Let  $s_1, \dots, s_b$  denote the children of  $t$  with navel  $K$  and define the matchgate  $\Delta = (G_{\leq s_1} \oplus_K \dots \oplus_K G_{\leq s_b}, K)$ . Recall that  $|K| \leq 3$  since  $\mathcal{T}$  is nice.
  2. Use Lemma 1 to compute  $f = \text{Sig}(\Delta)$  and use Fact 1 to obtain a planar matchgate  $\Phi$  on external vertices  $K$  with  $\text{Sig}(\Phi) = f$  and  $K$  on its outer face.
  3. Replace  $G_t$  by  $G_t \oplus_K \Phi$ , resulting in a planar graph: Planarity is obvious if  $|K| \leq 2$ . If  $|K| = 3$ , recall that  $K$  lies on the outer face of  $\Phi$ , and that  $K$  bounds a face in  $G_t$ . The union of such planar graphs preserves planarity.

After processing all attachment cliques, the graph  $G_t$  is planar and has  $\mathcal{O}(n_t)$  vertices. By Corollary 1, we have  $\text{Sig}(\Psi) = \text{Sig}(\Gamma_{\leq t})$  for  $\Psi = (G_t, K)$ , where  $K$  with  $|K| \leq 3$  is the navel of  $t$ . Compute  $\text{Sig}(\Psi)$  by Theorem 2 and Remark 3 in time  $\mathcal{O}(n_t^{1.5})$ .

By Theorem 3 and Remark 2, computing  $\mathcal{T}$  requires  $\mathcal{O}(n^4)$  time for general  $H$  or  $\mathcal{O}(n)$  time for  $H \in \{K_{3,3}, K_5\}$ . Processing  $\mathcal{T}$  requires time  $\mathcal{O}(|T| + \sum_{t \in T} n_t^{1.5})$ : At node  $t$ , we spend either  $2^{\mathcal{O}(c^2)}$  or  $\mathcal{O}(n_t^{1.5})$  time. Since  $\sum_{t \in T} n_t \in \mathcal{O}(n)$  by the size bound of Theorem 3, it follows that  $\sum_{t \in T} n_t^{1.5} \leq (\sum_{t \in T} n_t)^{1.5} \in \mathcal{O}(n^{1.5})$ . As  $|T| \in \mathcal{O}(n)$ , the overall runtime claims follow.

### 4 Conclusions and future work

We presented a polynomial-time algorithm for  $\text{PerfMatch}(G)$  on graphs  $G \in \mathcal{C}[H]$  when  $H$  is single-crossing. Since structural results about graphs in  $\mathcal{C}[H]$  for arbitrary (and not necessarily single-crossing) graphs  $H$  are known [15], it is natural to ask whether our approach can be extended to such graphs. We cautiously believe in an affirmative answer – in fact, Mingji Xia and the author made some progress towards a proof, but are still facing nontrivial obstacles.

## References

- [1] T. Asano. An approach to the subgraph homeomorphism problem. *Theor. Comp. Sci.*, 38(0):249–267, 1985.
- [2] H. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM J. Comput.*, 25(6):1305–1317, December 1996.
- [3] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*. Number 7 in Algorithms and Computation in Mathematics. Springer Verlag, 2000. 168 + xii pp.
- [4] E. Chambers and D. Eppstein. Flows in one-crossing-minor-free graphs. *J. Graph Algorithms Appl.*, 17(3):201–220, 2013.
- [5] E. Demaine, M. Hajiaghayi, N. Nishimura, P. Ragde, and D. Thilikos. Approximation algorithms for classes of graphs excluding single-crossing graphs as minors. *J. Comput. Syst. Sci.*, 69(2):166–195, 2004.
- [6] E. Demaine, M. Hajiaghayi, and D. Thilikos. 1.5-approximation for treewidth of graphs excluding a graph with one crossing as a minor. In *APPROX*, pages 67–80, 2002.
- [7] A. Galluccio and M. Loeb. On the theory of pfaffian orientations. I. Perfect matchings and permanents. *Electr. J. Comb.*, 6, 1999.
- [8] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.
- [9] P. Kasteleyn. Graph theory and crystal physics. In *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, 1967.
- [10] Igor Kriz and Robin Thomas. Clique-sums, tree-decompositions and compactness. *Discrete Mathematics*, 81(2):177 – 185, 1990.
- [11] R. Lipton, D. Rose, and R. Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16(2):346–358, 1979.
- [12] C. Little. An extension of Kasteleyn’s method of enumerating the 1-factors of planar graphs. In *Combinatorial Mathematics*, LNCS, pages 63–72. 1974.
- [13] B. Reed and Z. Li. Optimization and recognition for  $K_5$ -minor free graphs in linear time. In *LATIN 2008: Theoretical Informatics*, pages 206–215. 2008.
- [14] N. Robertson and P. Seymour. Excluding a graph with one crossing. In *Graph Structure Theory*, pages 669–676, 1991.
- [15] N. Robertson and P. Seymour. Graph minors. XVI. Excluding a non-planar graph. *Journal of Combinatorial Theory, Series B*, 89(1):43 – 76, 2003.
- [16] S. Straub, T. Thierauf, and F. Wagner. Counting the number of perfect matchings in  $K_5$ -free graphs. *Electronic Colloquium on Comp. Complexity (ECCC)*, 21(79), 2014.
- [17] H. Temperley and M. Fisher. Dimer problem in statistical mechanics - an exact result. *Philosophical Magazine*, 6(68):1061–1063, 1961.
- [18] L. Valiant. The complexity of computing the permanent. *Theor. C. Sci.*, pages 189–201, 1979.
- [19] L. Valiant. Holographic algorithms. *SIAM J. Comput.*, 37(5):1565–1594, 2008.
- [20] V. Vazirani. NC algorithms for computing the number of perfect matchings in  $K_{3,3}$ -free graphs and related problems. *Inf. Comput.*, 80(2):152–164, 1989.